

Análise da Complexidade Computacional

1. Selection Sort

Complexidade Computacional: $O(n^2)$

Análise: O Selection Sort é um algoritmo de ordenação simples, mas ineficiente para grandes conjuntos de dados devido à sua complexidade quadrática. Observando os tempos de execução, vemos que conforme o tamanho do vetor aumenta, o tempo de execução aumenta drasticamente. Para um vetor de tamanho 500.000, o tempo de execução é significativamente maior, refletindo a complexidade $O(n^2)$.

2. Merge Sort

Complexidade Computacional: $O(n \log n)$

Análise: O Merge Sort é mais eficiente que o Selection Sort, especialmente para grandes conjuntos de dados, devido à sua complexidade $O(n \log n)$. Os tempos de execução mostram um crescimento mais moderado comparado ao Selection Sort. Mesmo para um vetor de tamanho 500.000, o tempo de execução permanece em uma escala de milissegundos, mostrando a eficiência do algoritmo.

3. Counting Sort

Complexidade Computacional: $O(n + k)$ onde k é o intervalo de valores no vetor

Análise: O Counting Sort é extremamente eficiente para certos tipos de dados, especialmente quando o intervalo de valores (k) é limitado. Os tempos de execução são os menores entre os três algoritmos para todos os tamanhos de vetores fornecidos. Isto demonstra a eficiência do Counting Sort para grandes conjuntos de dados quando k é pequeno ou moderado.